

ER622185400

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

SECURITY ASSOCIATIONS FOR DEVICES

INVENTORS:

TREVOR FREEMAN

TIMOTHY MOORE

BERNARD ABOBA

ATTORNEY'S DOCKET NO.: MS1-1747US

## FIELD

**[0001]** The present invention relates to procuring trusted associations between at least two parties.

## BACKGROUND

**[0002]** Secret key encryption is implemented by a sending party encrypting data to be transmitted using a key, transmitting the key and the encrypted data to a receiving party either separately or together over a network connection, and the receiving party using a same key to decrypt the received data. A public key infrastructure (hereafter “PKI”) is implemented by the sending party encrypting data to be transmitted using a public key corresponding to the receiving party, transmitting the encrypted data to the receiving party over a network connection, and the receiving party using its private key to decrypt the received data. However, the trust required for sharing both secret keys for secret key encryption and public keys for PKI is being breached at an alarmingly increasing rate among distributed network appliances, particularly in constrained situations such as securing a boot among multiple appliances or nodes over a network.

## SUMMARY

**[0003]** Secure network associations among plural devices are described herein.

**[0004]** Generating symmetric keys among distributed appliances, includes generating public and private key values one at least one appliance, receiving a public key value from another appliance via an out-of-band third party mechanism, and generating a secret value as a function of the private key value corresponding to the local appliance and the public key value received from the other appliance.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0005]** The scope of the present invention will be apparent from the following detailed description, when taken in conjunction with the accompanying drawings, and such detailed description, while indicating embodiments of the invention, are given as illustrations only, since various changes and modifications will become apparent to those skilled in the art from the following detailed description, in which:

**[0006]** FIG. 1 shows a client/network system in accordance with example embodiments;

**[0007]** FIG. 2 shows an association of devices in accordance with an example embodiment;

**[0008]** FIG. 3 illustrates a processing flow in accordance with an example embodiment;

**[0009]** FIG. 4 illustrates another processing flow in accordance with an example embodiment; and

**[0010]** FIG. 5 illustrates a general computer network environment which can be used to implement the techniques described herein.

## DETAILED DESCRIPTION

**[0011]** In the example network environment of FIG. 1, multiple client computing devices 105, 110, 115, and 120, which are also referred to as client devices, are coupled to each other and to at least one server device 125 via network 100. Network 100 is intended to represent any of a variety of conventional network topologies and types, which may include wired and/or wireless networks. Network 100 may further utilize any of a variety of conventional network protocols, including public and/or proprietary

protocols. Network 100 may include, for example, the Internet as well as possibly at least portions of one or more local area networks (LANs).

**[0012]** Client device 105 may include any of a variety of conventional computing devices, including a desktop personal computer (PC), workstations, mainframe computers, Internet appliances, and gaming consoles. Further client devices associated with network 100 may include personal digital assistant (PDA) 110, laptop computer 115, and cellular telephone 120, etc., which may be in communication with network 100 by a wired and/or wireless link. Further still, one or more of client devices 105, 110, 115, and 120 may include the same types of devices, or alternatively different types of devices.

**[0013]** Server device 125 may provide any of a variety of data and/or functionality to computing devices 105, 110, 115, and 120. The data may be publicly available or alternatively restricted, *e.g.*, restricted to only certain users or available only if the appropriate fee is paid, *etc.* Server device 125 is at least one of a network server, an application server, a web blade, or any combination thereof. Server device 125 is any device that is the source of content, and client devices 105, 110, 115, and 120 include any devices that receive such content.

**[0014]** At data source 130 or 135, software programs, including operating systems and applications, are prepared for and/or provided to any one of server device 125 or client devices 105, 110, 115, and 120 for loading and/or execution. For the sake of consistency, the discussion hereafter refers to “applications” which encompass anyone of, at least, operating systems, programs, and applications, as known in the art, either singularly or in combination.

**[0015]** Secure associations between any of client devices 105, 110, 115, 120, server device 125, and data sources 130 and 135 is described herein with reference to the example embodiment of FIG. 2.

**[0016]** In particular, the example embodiment of FIG. 2 shows how a trust relationship is established between device 205 and device 230, with devices 205 and 230 representing any one of client devices 105, 110, 115, 120, server device 125, and data sources 130 and 135. Such devices may alternatively be regarded as network nodes. Trust relationships, or secure associations, in accordance with the example embodiments described herein, may be implemented by establishing a shared secret without requiring a high degree of trust for the dissemination of information between devices. The trust relationship may be established for the purposes that include, but are not limited to, exchanging information, *e.g.*, e-mail, or for remotely loading an operating system (OS) from one node to another.

**[0017]** According to the example embodiment of FIG. 2, a trust relationship between devices 205 and 230 is not established over network 100, over which devices 205 and 230 are communicatively coupled, but rather using a third-party out-of-band, *i.e.*, asynchronous, entity. Such entity, which will be referred to hereafter as “out-of-band mechanism” 245 may include, but is not limited to, any one of: a serial cable, a USB cable, an infrared-connection, a personal digital assistant (PDA), a flash memory, a memory stick, a barcode, and a smartcard. Out-of-band mechanism 245 may be program-driven or require user-intervention. Serial cable, USB cable, and infrared-compatible examples of out-of-band mechanism 245 may be alternatively program-driven or user-implemented. Peripheral devices such as a PDA, a flash memory, a memory stick, a barcode, and a smartcard serving as out-of-band mechanism 245 require

physical transfer from one device to another, and therefore require user-intervention in order for data to be downloaded to a destination device.

**[0018]** According to a first embodiment, symmetric keys are established on at least devices 205 and 230 using the Diffie-Hellman cryptographic protocol. In particular, generator 215 on device 205 and generator 235 on device 230 each produce a local public/private key pair for the respective devices. The public key values generated on devices 205 and 230 are exchanged via out-of-band mechanism 245. Thus, having imported the public key value generated on the other device via out-of-band mechanism 245, devices 205 and 230 are able to produce a shared secret as a function of the imported public key value and the local private key value by executing a Diffie-Hellman computation, which is known in the art and is therefore not described in detail here.

**[0019]** That is, shared secret generator 225 on device 205 produces a Diffie-Hellman shared secret as a function of the private key value produced by generator 215 and the public key value imported from device 230 via out-of-band mechanism 245. Further, shared secret generator 240 on device 230 produces a Diffie-Hellman shared secret as a function of the private key value produced by generator 235 and the public key value imported from device 205 via out-of-band mechanism 245. As is known in the art with regard to the Diffie-Hellman cryptographic technique, by exchanging public keys, the shared secret key values generated on devices 205 and 230 are the same, *i.e.*, symmetric, but neither device is required to export either a private key value or the shared secret value over a network. Rather, only a public key value is transmitted from one device to another, and that over an out-of-band mechanism, requiring only a low level of trust.

[0020] The Diffie-Hellman secret value generated at each of the devices is used for encryption/decryption or other known authentication purposes.

[0021] The embodiment of FIG. 2 may also establish a secret value to be shared among at least devices 205 and 230 using the Rivest-Shamir-Adleman (hereafter “RSA”) cryptographic protocol. According to such example embodiment, a secret value is produced on one of devices 205 and 230, and exported to the other node via out-of-band mechanism 245 while protected by the public key value corresponding to the destination device.

[0022] Specifically, to implement the RSA protocol, devices 205 and 230 are to produce a public key value, though only one of the devices is needed to produce a private key value. The private key value is produced on the device that is to generate the secret value to be shared. The description of the example embodiment continues assuming that generator 215 on device 205 and generator 235 on device 230 each produce a local public/private key pair for the respective devices, though such example is not limiting.

[0023] Assuming that device 205 is to generate the secret value to be shared, the public key value generated by generator 235 on device 230 is imported to device 205 via out-of-band mechanism 245. Shared secret generator 225 on device 205 produces an RSA secret value as a function of the local private key value produced by generator 215 and the public key value imported from device 230 via out-of-band mechanism 245. As is known in the art with regard to the RSA cryptographic technique, the secret value is then shared with device 230 by encrypting the secret value using the imported public key value, *i.e.*, the public key value imported to device 205 from device 230 via out-of-band mechanism 245. Out-of-band mechanism 245 is then utilized again to export the secret value to device 230, with the secret value protected by the public key value of device

230. Thus, devices 205 and 230 benefit from sharing the secret value, which is then used for encryption/decryption or other known authentication purposes.

**[0024]** An example implementation for the embodiments described herein includes providing a secure boot over a network wherein boot loader code resides on the firmware of a network appliance. A description of such an example is hereby described with reference to the example embodiment of FIG. 2 and the example processing flow shown in FIG. 3. This description is exemplary only, and is not intended to be limiting in any manner. Further, the processing of FIG. 3 is not intended to be limiting in terms of order of actions taken since, as will become evident from the description, any sequence in the processing may be varied.

**[0025]** The processing of FIG. 3 is described with regard to the example of FIG. 2 whereby device 205 represents a RADIUS client and device 230 represents a RADIUS server. Remote Authentication Dial-In User Service (RADIUS) is an authentication service that allows a service infrastructure to maintain user profiles in a centralized database residing on an authentication server, and the profiles are then accessed by remote access servers.

**[0026]** When RADIUS client 205 attempts to remotely load an OS, the profile of which is stored on RADIUS server 230, public/private key values are generated 305 by generator 215 on RADIUS client 205. The public key value generated on RADIUS client 205 is exported 310 to RADIUS server 230 via out-of-band mechanism 245.

**[0027]** Public/private key values are also generated 305 on RADIUS server 230 by generator 235, either simultaneously with the generation of the public/private key values on RADIUS client 205 or, more likely, in response to receiving the public key value from RADIUS client 205 at RADIUS server 230. The public key value generated



on RADIUS server 230 is exported 310 to RADIUS client 205 via out-of-band mechanism 245.

**[0028]** Generator 240 on RADIUS server 230 generates 315 a shared secret value by executing a Diffie-Hellman computation, with the shared secret being computed as a function of the private key value generated by generator 235 and the public key value imported from RADIUS client 205 via out-of-band mechanism 245.

**[0029]** Similarly, generator 225 on RADIUS client 205 generates 315 a shared secret value by executing a Diffie-Hellman computation, with the shared secret being computed as a function of the private key value generated by generator 215 and the public key value imported from RADIUS server 230 via out-of-band mechanism 245.

**[0030]** The secret values generated by generator 225 and generator 240 are symmetric, in accordance with the fundamental principals of the Diffie-Hellman encryption technique.

**[0031]** Alternatively, the secure boot of an OS from RADIUS server 230 on RADIUS client 205 can also be implemented using an RSA encryption technique. The description of such embodiment is described with reference to the examples of FIGS. 2 and 4. Further, the example is described under the assumption that RADIUS client 205 initiates the establishment of the trust relationship between RADIUS client 205 and RADIUS server 230, although such example is not limiting. It is noted that the depiction and number of the blocks of FIGS. 3 and 4 do not imply a strict order. Rather, the ordering described is by way of example only.

**[0032]** Therefore, by FIG. 4, when RADIUS client 205 attempts to remotely load an OS, the profile of which is stored on RADIUS server 230, public/private key values are generated 405 by generator 215 on RADIUS client 205. The public key value

generated on RADIUS client 205 is exported 410 to RADIUS server 230 via out-of-band mechanism 245.

**[0033]** Public/private key values are also generated 405 on RADIUS server 230 by generator 235, either simultaneously with the generation of the public/private key values on RADIUS client 205 or in response to receiving the public key value from RADIUS client 205 at RADIUS server 230.

**[0034]** Generator 240 on RADIUS server 230 generates 415 a shared secret value by executing an RSA computation, with the shared secret being computed as a function of the private key value generated by generator 235 and the public key value imported from RADIUS client 205 via out-of-band mechanism 245.

**[0035]** The secret value generated on RADIUS server 230 is encoded to be protected by the public key value imported from RADIUS client 205, and exported 420 to RADIUS client 205 via out-of-band mechanism 245.

**[0036]** The techniques described above may also be utilized to establish a trust relationship between any of devices 105, 110, 115, 120, server 125, and data sources 130 and 135, shown in FIG. 1, for alternative purposes including the dissemination of data, *e.g.* e-mail, establishing a secure link for videoconferencing, *etc.* That is, the techniques described above utilize an out-of-band entity to reduce the trust required of any network node desiring a trust relationship with at least one other node, regardless of the purpose for such association.

**[0037]** Further, with regard to the example embodiments described herein, it is noted that means and methods for the generation of public/private key values are known in the art, and therefore are not presently described in detail. Similarly, encryption protocols including, but not limited to, the Diffie-Hellman protocol and the RSA

protocol, which may be utilized in correspondence with the example embodiments described herein are also known, and therefore a description of such protocols is not provided.

**[0038]** FIG. 5 illustrates a general computer environment 500, which can be used to implement the techniques described herein. The computer environment 500 is only one example of a computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the computer and network architectures. Neither should the computer environment 500 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the example computer environment 500.

**[0039]** Computer environment 500 includes a general-purpose computing device in the form of a computer 502. The components of computer 502 can include, but are not limited to, one or more processors or processing units 504, system memory 506, and system bus 508 that couples various system components including processor 504 to system memory 506.

**[0040]** System bus 508 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, such architectures can include an Industry Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA) local bus, a Peripheral Component Interconnects (PCI) bus also known as a Mezzanine bus, a PCI Express bus, a Universal Serial Bus (USB), a Secure Digital (SD) bus, or an IEEE 1394, i.e., FireWire, bus.

**[0041]** Computer 502 may include a variety of computer readable media. Such media can be any available media that is accessible by computer 502 and includes both volatile and non-volatile media, removable and non-removable media.

**[0042]** System memory 506 includes computer readable media in the form of volatile memory, such as random access memory (RAM) 510; and/or non-volatile memory, such as read only memory (ROM) 512 or flash RAM. Basic input/output system (BIOS) 514, containing the basic routines that help to transfer information between elements within computer 502, such as during start-up, is stored in ROM 512 or flash RAM. RAM 510 typically contains data and/or program modules that are immediately accessible to and/or presently operated on by processing unit 504.

**[0043]** Computer 502 may also include other removable/non-removable, volatile/non-volatile computer storage media. By way of example, FIG. 5 illustrates hard disk drive 516 for reading from and writing to a non-removable, non-volatile magnetic media (not shown), magnetic disk drive 518 for reading from and writing to removable, non-volatile magnetic disk 520 (e.g., a “floppy disk”), and optical disk drive 522 for reading from and/or writing to a removable, non-volatile optical disk 524 such as a CD-ROM, DVD-ROM, or other optical media. Hard disk drive 516, magnetic disk drive 518, and optical disk drive 522 are each connected to system bus 508 by one or more data media interfaces 525. Alternatively, hard disk drive 516, magnetic disk drive 518, and optical disk drive 522 can be connected to the system bus 508 by one or more interfaces (not shown).

**[0044]** The disk drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules, and other data for computer 502. Although the example illustrates a hard disk 516,

removable magnetic disk 520, and removable optical disk 524, it is appreciated that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes or other magnetic storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or other optical storage, random access memories (RAM), read only memories (ROM), electrically erasable programmable read-only memory (EEPROM), and the like, can also be utilized to implement the example computing system and environment.

**[0045]** Any number of program modules can be stored on hard disk 516, magnetic disk 520, optical disk 524, ROM 512, and/or RAM 510, including by way of example, operating system 526, one or more application programs 528, other program modules 530, and program data 532. Each of such operating system 526, one or more application programs 528, other program modules 530, and program data 532 (or some combination thereof) may implement all or part of the resident components that support the distributed file system.

**[0046]** A user can enter commands and information into computer 502 via input devices such as keyboard 534 and a pointing device 536 (e.g., a “mouse”). Other input devices 538 (not shown specifically) may include a microphone, joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and other input devices are connected to processing unit 504 via input/output interfaces 540 that are coupled to system bus 508, but may be connected by other interface and bus structures, such as a parallel port, game port, or a universal serial bus (USB).

**[0047]** Monitor 542 or other type of display device can also be connected to the system bus 508 via an interface, such as video adapter 544. In addition to monitor 542,

other output peripheral devices can include components such as speakers (not shown) and printer 546 which can be connected to computer 502 via I/O interfaces 540.

**[0048]** Computer 502 can operate in a networked environment using logical connections to one or more remote computers, such as remote computing device 548. By way of example, remote computing device 548 can be a PC, portable computer, a server, a router, a network computer, a peer device or other common network node, and the like. Remote computing device 548 is illustrated as a portable computer that can include many or all of the elements and features described herein relative to computer 502. Alternatively, computer 502 can operate in a non-networked environment as well.

**[0049]** Logical connections between computer 502 and remote computer 548 are depicted as a local area network (LAN) 550 and a general wide area network (WAN) 552. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

**[0050]** When implemented in a LAN networking environment, computer 502 is connected to local network 550 via network interface or adapter 554. When implemented in a WAN networking environment, computer 502 typically includes modem 556 or other means for establishing communications over wide network 552. Modem 556, which can be internal or external to computer 502, can be connected to system bus 508 via I/O interfaces 540 or other appropriate mechanisms. It is to be appreciated that the illustrated network connections are examples and that other means of establishing at least one communication link between computers 502 and 548 can be employed.

**[0051]** In a networked environment, such as that illustrated with computing environment 500, program modules depicted relative to computer 502, or portions thereof, may be stored in a remote memory storage device. By way of example, remote

application programs 558 reside on a memory device of remote computer 548. For purposes of illustration, applications or programs and other executable program components such as the operating system are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of computing device 502, and are executed by at least one data processor of the computer.

**[0052]** Various modules and techniques may be described herein in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. for performing particular tasks or implement particular abstract data types. Typically, the functionality of the program modules may be combined or distributed as desired in various embodiments.

**[0053]** An implementation of these modules and techniques may be stored on or transmitted across some form of computer readable media. Computer readable media can be any available media that can be accessed by a computer. By way of example, and not limitation, computer readable media may comprise “computer storage media” and “communications media.”

**[0054]** “Computer storage media” includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage

or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computer.

**[0055]** “Communication media” typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as carrier wave or other transport mechanism. Communication media also includes any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. As a non-limiting example only, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above are also included within the scope of computer readable media.

**[0056]** Reference has been made throughout this specification to “one embodiment,” “an embodiment,” or “an example embodiment” meaning that a particular described feature, structure, or characteristic is included in at least one embodiment of the present invention. Thus, usage of such phrases may refer to more than just one embodiment. Furthermore, the described features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

**[0057]** One skilled in the relevant art may recognize, however, that the invention may be practiced without one or more of the specific details, or with other methods, resources, materials, etc. In other instances, well known structures, resources, or operations have not been shown or described in detail merely to avoid obscuring aspects of the invention.

**[0058]** While example embodiments and applications of the present invention have been illustrated and described, it is to be understood that the invention is not limited



to the precise configuration and resources described above. Various modifications, changes, and variations apparent to those skilled in the art may be made in the arrangement, operation, and details of the methods and systems of the present invention disclosed herein without departing from the scope of the claimed invention.